

Project Management Efficiency using Soft computing and Risk Analysis

Vinay Kumar Nassa
Research Scholar
Singhania University Pacheri Bari,
Jhunjhunu(Raj.),India

Sri Krishan Yadav
PhD, Prof. & HOD (Applied Sciences)
Dronacharya College of Engineering, Gurgaon-
123506

ABSTRACT

A software project in general involves the presence of risk and hence risk analysis is required.. Most software project managers do it informally and superficially, if they do it at all. The time spent identifying, analyzing, and managing risk pays itself back in many ways: less upheaval during the project, a greater ability to track and control a project, and the confidence that comes with planning for problems before they occur. The software project planner must estimate three things before a project begins: how long it will take, how much effort will be required, and how many people will be involved. In addition, the planner must predict the resources (hardware and software) that will be required and the risk involved.

The basic objective of the paper shall revolve around the concepts on project management, soft computing and risk analysis techniques. The objective of this paper is to present an approach for creating a robust risks classifications and measurement system. This paper quantify the project management efficiency (PME) using a Soft computing tool based fuzzy logic system (SFLS) employing risk analysis. This approach indeed provides simplification and reduces the effort and time to perform an analysis for project selection. The evaluation of PME can serve for project managers and for project organization as indicator for the level of achievement of project management objectives. PME may help in the evaluation of performance of project teams.

Keywords

Project management efficiency (PME); Softcomputing;Risk Analysis; Fuzzy sets; Project time delay; Project time delay gradient.

1. INTRODUCTION

Effective software project management focuses on the four P's: people, product, process, and project. The order is not arbitrary. Four P's have a substantial influence on software project management—people, product, process, and project. A project may be viewed as “the entire process required to produce a new product, new plant, new system, or other specific results” [12].In this fiercely competitive world, project organizations are forced to look for scientific tool that assist them in the evaluation of their projects. The project management team is responsible for producing the project output and hence the project management team must be constantly aware of the project goal, project purpose and project management efficiency. The manager who forgets that software engineering work is an intensely human endeavor will never have success in project management. A manager who fails to encourage comprehensive customer

communication early in the evolution of a project risks building an elegant solution for the wrong problem. The manager who pays little attention to the process runs the risk of inserting competent technical methods and tools into a vacuum.

No doubt, for the successful completion and maintance of software projects, there is a need for effective project managers. The long proclaimed ineffectiveness of software projects to maintain their schedule, cost, and quality, continues to plague most development projects [6, 10,5]. It has been observed that over half of all software development projects are considered a failure with respect to their cost and schedule [6]. The software crisis must be addressed and, to the extent possible, resolved. To do so require more accurate schedule and cost estimates, better products, and higher productivity. All these can be achieved through more effective software management.

In the field of software development effort on the no of risk management techniques is done in the past decade. Risk is the possibility of suffering harm or loss which causes danger. In a technology driven area of study, software project risk management has been relevant over time and indicates the importance and complexity of the issues they are addressing. Since risk is uncertain, project managers must somehow determine the impact the risks will have on the project. Project mangers try to assess risk using exact values and fail. Since risk cannot quantified in straight, crisp terms it must be taken and analyzed as a distribution.

The software development industry is probably one of the most risky of industries at the present moment. In order to mature the development process in terms of cost, effort and time, an extensive effort is now being made by many of the software industries [6]. Risks factors are present throughout the whole development process and these can negatively affect the project. These software risk factors are of concern since there has been a large amount of software disasters occurring recently. A recent survey of 600 firms indicated that 35 percent of them had at least one runaway software project [2].

2. LITERATURE SURVEY

Software plays an important role more or less in each and every aspect of human life. But, at the same movement, it has been the most troubling technology of the 20th century [8]. Generally, it has been observed that most of the software projects have the highest probability of being cancelled or delayed of any known business activity. Software projects often display excessive error densities and low levels of reliability on deployment. Because of its highly unreliable and

disastrous nature, it is achieving a very bad public reputation. A careful implementation of risk analysis and mitigation may reduce the probability of major disasters [1]. Extensive research has been done to develop sophisticated tools that can analyze and provide accurate information for the choice of investments and projects. Many of these researchers though, have faced the dilemma that much of their data is plagued by uncertainty, vagueness and approximation. This uncertainty, vagueness, and approximation can be seen mostly in the area of risk analysis where risk is considered a black box in which the few who have ventured to take it into consideration and analyze it have failed to come back out of the box with a clear and understandable analysis, and therefore the majority of project managers have been afraid to even delve into this area.

Risk is the possibility of suffering harm or loss which causes danger. In a technology driven area of study, software project risk management has become relevant over time and indicates the importance and complexity of the issue they are addressing. Not only has the issue of software risk management been researched for more than two decades now, but, as years go by, it appears to have attracted the attention of an increasing number of researchers [7]. It is evident from the literature survey that several aspects of software project risk management have been studied using a variety of approaches. It has been observed that the risk management software project has been an area of research since 1970's and many of academicians and industry practitioners are producing their views on the same [9]. In 1999, a survey conducted on 34 empirical studies on software risk management published during 1978-1999, reveals that the area is enrich in terms of approaches, ranging from action research and case studies to survey and laboratory experiments [11].

During the development of sophisticated tools, many of these researchers found that their data is plagued by uncertainty, vagueness and approximations. This uncertainty, vagueness and approximations is important in the area of risk analysis, which when applied for analysis, did not result into clear and understandable analysis.

Many research projects have been analyzed using risk, but a research project representing the fuzzy logic in a complex project full of risk is represented in the research . This research creates 'a hierarchical risk ' breakdown structure representation used to develop a formal model of qualitative risk assesment. It also represents a methodology for evaluating the risk exposure considering the consequences in terms of time, cost, quality and safety performance measure of a project based on fuzzy estimate of the risk components. This research provides a very detailed decomposition and analysis of risks, taking into consideration uncertainty. Yet the focus is not in project management efficiency, but rather its focus is to assess risk. Since the work in this paper mainly considers the project management efficiency in the area of software development, it is very important to determine the risks that plague this industry. A list of risks is provided as a basis work breakdown structure [4], by Software Engineering Institute. This list is taxonomy of risks divided into families, groups and subgroups giving the broad description of risks that plague the software development industry. The cause and effect diagram which allows project managers to have a better comprehension of the effects that risks have upon a project is represented in this [2]. This provides the need for an analysis that can take into consideration the uncertainty that effects the software development industry ,since crisp values cannot be given to risks.

3. IDENTIFICATION AND PROBLEM DEFINITION

Risk is the possibility of suffering harm or loss which describes the impact to the project in form of diminished quality of end product, project failure.

Dangers in the new and emerging field of software engineering must often be learned without the benefit of lifelong exposure. An approach involving studying the experiences of successful project managers as well as keeping up with the leading practitioners and researchers in the area is required. The major objective of software risk management include indentifying, addressing, and eliminating potential elements of risk before they become either threats to successful software operation or major sources of software network.

It is proposed that Soft computing techniques particularly fuzzy logic technique can help to overcome some of these difficulties by representing the imprecision in inputs and outputs as well as providing more expert knowledge based approach to model building. One of the important benefits of fuzzy logic for software engineering project management is the flexibility in terms of the types of input and output variables. The fuzzy logic is composed of fuzzy sets, which are provided by a mathematical definition rising from the concept of degree of membership, which increases the number of possibilities that can be subject to research.

4. PROPOSED STUDY APPROACH

In a technology driven area of study, software project risk management has become relevant over time and indicates the importance and complexity of the issue they are addressing. This thesis required an intensive research work into the risk analysis and software development. Software development industry is plagued with risk due to long proclaimed ineffectiveness of software development projects to maintain their schedule, cost and quality. The problem chosen to be due to the reason that many project managers are not willing to take into consideration the risk because risk analysis techniques are either too complex or they are inefficient. The objective of this work is too simplify the project manager's job and allow him or her to determine the risks that impact the project by just using his or her expertise naturally (expressions) as opposed to forced (values).

The methodology undertaken to develop the proposed approach is represented in flowchart.

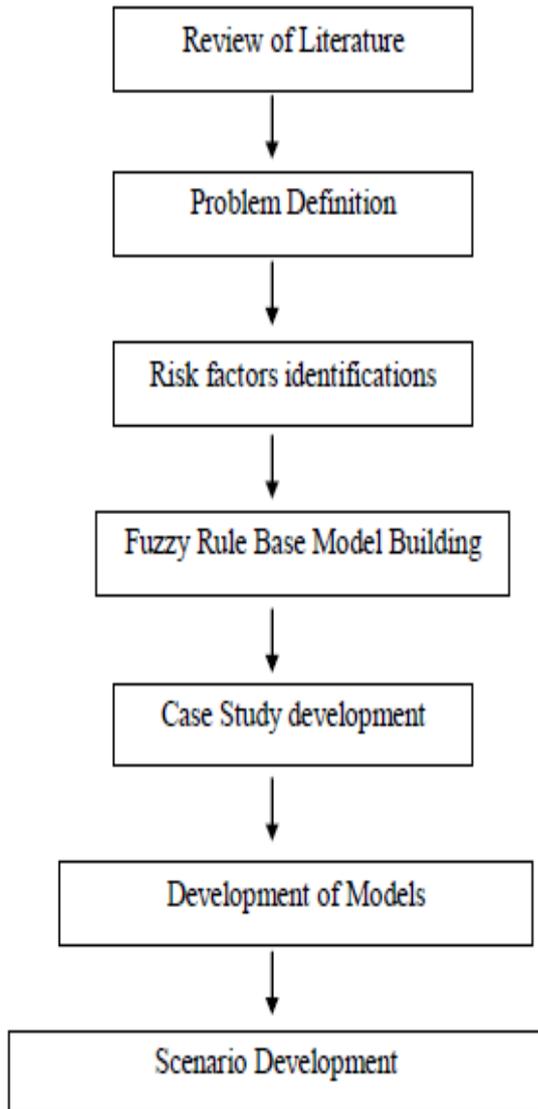


Figure 1: Study Approach

5. SOFT COMPUTING PARADIGM

Prior to 1994 when Zadeh first defined “soft computing“, the currently-handled concepts used to be referred to in an isolated way, whereby each was spoken of individually with an indication of the use of fuzzy methodologies. Although the idea of establishing the area of soft computing dates back to 1990, it was in that Zadeh established the definition of soft computing in the following terms:

“Basically, soft computing is not a homogeneous body of concepts and techniques. Rather, it is a partnership of distinct methods that in one way or another conform to its guiding principle.” At this juncture, the dominant aim of soft computing is to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solutions cost. The principal constituents of soft computing are fuzzy logic, neurocomputing, and probabilistic reasoning, with the latter subsuming genetic algorithms, belief networks, chaotic systems, and parts of learning theory.

It is therefore clear that rather than a precise definition for soft computing, it is instead defined by extension, by means of different concepts and techniques which attempt to overcome the difficulties which arise in real problems which occur in a world which is imprecise, uncertain and difficult to categorize.

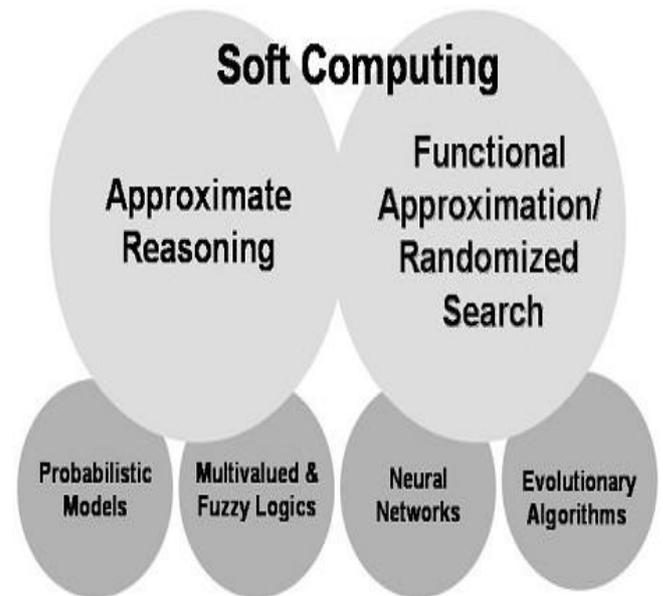


Figure 2 : Soft Computing Components

5.1 Fuzzy Logic and Fuzzy Set

Fuzzy logic is a relatively new technique (first appeared in 1970s) for solving engineering control problems. This technique can be easily used to implement systems ranging from simple, small or even embedded up to large. The key idea of fuzzy logic is that it uses a simple and easy way in order to get the output(s) from the input(s), actually the outputs are related to the inputs using if-statements and this is the secret behind the easiness of this technique. The most fascinating thing about Fuzzy logic is that it accepts the uncertainties that are inherited in the realistic inputs and it deals with these uncertainties in such away their affect is negligible and thus resulting in a precise outputs.

Although fuzzy control was not the first engineering application of fuzzy logic, it was the first application that drew huge attention to the practical potential of fuzzy set theory. It uses many elements of fuzzy logic to define a rule-base for the controller.

Fuzzy set theory (FST) is a generalization of the ordinary set theory. A fuzzy set is a set whose elements belong to the set with some degree of membership μ .

Let X be a collection of objects. It is called universe of discourse. A fuzzy set $A \in X$ is characterized by membership function $\mu_A(x)$ represents the degree of membership, Degree of membership maps each element between 0 and 1.

It is defined as :

$$A = \{(x, \mu_A(x)); x \in X\} \dots\dots\dots 1$$

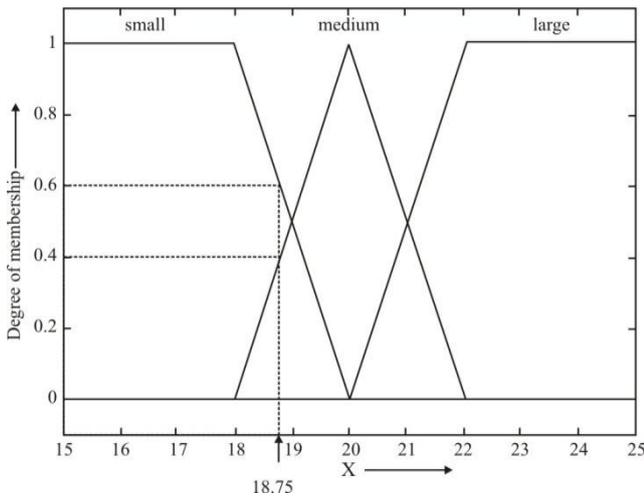


Figure 3: Fuzzy logic membership function

6. RISK IDENTIFICATION

The risks taken into consideration were extracted from taxonomy of risks developed for the software development industry. This taxonomy of risks is an extremely general account of the risk factors that affect the software industry. But since this taxonomy is extremely general and it is not portrayed as a cause and effect, a more specific case study ensues, where all the risks are portrayed in a causal and cognitive map “as a means of making visible the perceptions of project stakeholders with regard to the causes of shortcomings in completed IS/IT development projects.” [2]

This causal and cognitive map is presented in Figure 4.

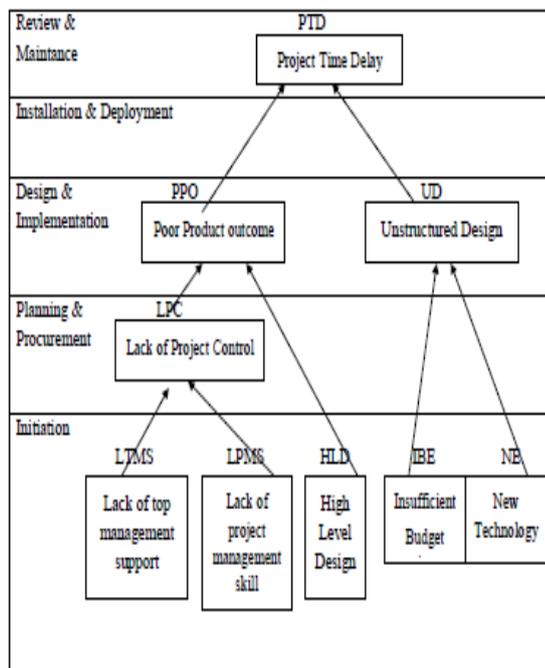


Figure 4: Causal and cognitive map

The risks are defined as such

1. **Lack of top management support** - Risk due to non commitment, sponsorship, an end goals perception, funding and proper training.
2. **Lack of project management skills** – Risk due to the improper skills of the project manger or team, most likely due to experience.
3. **High level design** – Risk due to complex project, that the design is therefore very difficult and tedious.
4. **Insufficient budget estimates** – Risk due to the fact that the budget set for the project is not proper (low) and it does not meet even the basic necessities for the completion of project.
5. **New technology** - Risk due to new technology used to develop a project since it requires learning & familiarization to staff, therefore will have problems developing the project on time or with good quality.
6. **Lack of project control** – Risk that the project deviates from original plan and follows in the undesirable direction.
7. **Unstructured design** - Risk that the software will be unorganized and difficult to use or understand.
8. **Poor product outcome** - Risk that the software does what it is supported to do but is not user friendly and self explanatory.

The consequences of all their risks is

- Project Time Delay - Risk that the project takes the time as a percentage of planned duration i.e. overrun/order run.

7. ALGORITHM

The research work has the purpose of presenting a new tool and a new technique to the software development technology. The algorithm presented in the work is composed of three major models: project time delay (PTD), project time delay gradient (PTDG) and project management efficiency (PME).

The algorithm work that the identified (8 risks) combine and interact with each other to build sub models which ultimately contribute to the project time delay (PTD) and project time delay gradient (PTDG). Therefore the two models are created, one to model the interrelationships of the risks contributing to be project time delay and other one to model the interrelationships of the risks contributions the gradient due to project time delay. These two models return a single value each, the first return the project time delay and second one returns the gradient of project time delay. The third model is being created as well which determines the relationship between two models and gives an output for the project management efficiency (PME) .

The input and output variables, membership functions and rules come from complete knowledge of process and expert knowledge. The various risk facts are abbreviated and listed below:

- LTMS - Lack of top management support
- LPMS- Lack of project Management skill
- LPC - Lack of project control

HLD-	High level design
PPO	-Poor product outcome
IBE	-Insufficient Budget estimate
NE	-New Technology
UD	-Unstructured design
PTD	-Project time delay

The first function that our fuzzy system is to perform is that of fuzzification. These to do this, we must define what is Low, Medium, High and Very High, in terms of risks and project management efficiency.

To define this commonly used techniques one proposed by Wang and Mendel Approach. In this approach the input and output variable are divided into various regions using some clustering technique namely fuzzy C-means clustering (FCM) and these are represented below.

The rules developed are shown in fuzzy associative map (FAM) for various risk factors.

Table 1: Input /Output Variable Representation

Variable	Risk (0-100)	Risk Gradient (0-100)	Output Project Management efficiency (0-100)
Low(L)	0-30	0-25	0-20
Medium(M)	20-50	20-45	15-55
High(H)	45-75	40-70	45-80
Very High(V _H)	65-100	65-80	75-100

Table 2: Rule base representation

		LPC			
		LPMS			
LTMS		L	M	H	V _H
	L	L	L	M	M
	M	L	M	M	H
	H	M	M	H	H
	V _H	M	H	H	V _H

8. SCENARIO

A number of scenarios have been introduced to represent the developed Fuzzy rule base Model and evaluated using Matlab scripts, Rule viewer and 3Dgraphs for the Project Time Delay(PTD) and Project Time Delay Gradient(PTDG) to determine the Project Management Efficiency (PME).

8.1 Scenario I (Low Value of Risk)

Table 3: Low Value of Risk

S.No	Risk Acronym	Description	Risk Value	Gradient Value
1	LTMS	Lack of top management support	08	09
2	LPMS	Lack of project management skill	18	20
3	LPC	Lack of project control	12.1	12.7
4	HLD	High level design	20	06
5	PPO	Poor project outcome	12.5	12.3
6	IBE	Insufficient budget estimate	09	19
7	NE	New Technology	12	07
8	UD	Unstructured design	11	12.4

1. Project Time Delay (PTD): : 10.8
2. Project Time Delay Gradient(PTDG): 11
3. Project Management Efficiency (PME): 89.3

8.2 Scenario II (Medium Value of Risk)

Table 4: Medium Value of Risk

S.No	Risk Acronym	Description	Risk Value	Gradient Value
1	LTMS	Lack of top management support	42	37
2	LPMS	Lack of project management skill	39	36
3	LPC	Lack of project control	35.2	32

4	HLD	High level design	43	34
5	PPO	Poor project outcome	35.2	32.5
6	IBE	Insufficient budget estimate	42	37
7	NE	New Technology	38	40
8	UD	Unstructured design	35.2	32.5

1. **Project Time Delay (PTD): : 35.3**
2. **Project Time Delay Gradient(PTDG): 32.5**
3. **Project Management Efficiency (PME): 35.2**

9. CONCLUSION

The model created in this work comprises of Risk Analysis approach to determine project management efficiency (PME) based upon the Soft computing technique. This work provides a contribution to the area of project management since it gives the project manager a simple yet powerful tool that helps him/her to quickly accept or discard the project before any complex analysis like cash flow or even ROI (Return on Investment). The algorithm developed in this thesis is based upon fuzzy logic, giving it the ability to solve complex problems plagued with uncertainty and vagueness. Since the software development industry is developing at extremely fast rate, there are lots of risks involved that can affect the outcome of projects and industry is still not completely adopt at dealing with risk. These risks are uncertain in nature and cannot be analyzed by traditional techniques.

This algorithm allows the user to input values related to risk occurrence, their gradients and return a value for project management efficiency (PME). The simplicity and easiest approach of this work in the risk analysis area provides a new dimension in Software project development.

9. REFERENCES

- [1] A.D. Buttingieg, "Risk Management in a Software Development Life Cycle", Web reference. Available at: <http://www.cis.um.edu.mt/~ABOUT/#Section%202>.
- [2] Al-Shehab, Abdullah J. Hughes, Robert T. and Winstanley, Graham "Modelling Risks in IS/IT Projects through Causal and Cognitive Mapping". Brighton, UK. The Electronic Journal of Information Systems Evaluation, Vol. 8, Iss. 1, pgs 1-10, 2005.
- [3] Boehm, Barry W. "Software Risk Management: Principles and Practices". *IEEE Software*, vol. 8, no. 1, pgs. 32-41, Jan/Feb, 1991
- [4] Carr, Marvin J.; Konda, Suresh L.; Monarch, Ira; Ulrich, Carol F.; Walker, Clay F. "Taxonomy-Based Risk Identification". Pittsburgh., PA: Software Engineering Institute, Carnegie Mellon University, June 1993.
- [5] D.H Kitson and S. Masters, "An Analysis of SEI Software Process Results 1987-1991.", Proc. 15th International Conference on Software Engineering, pp 68-77, 1993.ingieg
- [6] D.Merrill, "Software Development Project Managers with a Software Project Simulator" Master of Science Thesis Proposal, Department of Computer Science and Engineering Arizons State University Training Feb 4, 1996.
- [7] H.Barki, S. Rivard and J. Talbot, "An Integrative Contingency Model of Software Project Risk Mangement ", Journal of Management Information System/ Springer Vol. 17, No. 4, pp 37-69, 2001.
- [8] Jones, Capers, "Minimizing the Risks of Software ", May 1998
- [9] M.Keil, P.E. Cule, K. Lyytinen and R.C Schmidt, "A framework for identifying software project risks", Comm. Of ACM, 41, 11, 76-83, 1998.
- [10] B. W. Boehm, "Software Engineering Economics", Prentice-Hall, Enlewood Cliffs, New Jersey, 1981.
- [11] J.Ropponen, "Software Risk Management" Foundation", Principles and Empirical Findings. Jyvaskyla: Jyvaskyla University Printing House, 1999.
- [12] R.D. Archibald, Managing High-Technology Programs and Projects, John Wiley, New York, 1976.